



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO NEZAHUALCÓYOTL

LICENCIATURA EN INGENIERÍA EN SISTEMAS INTELIGENTES

**MANUAL PARA PRÁCTICAS DEL
LABORATORIO DE CÓMPUTO**

ASIGNATURA:

SISTEMAS OPERATIVOS

ELABORARÓN:

M. en C. YAROSLAF AARÓN ALBARRÁN FERNÁNDEZ

DRA. DORICELA GUTIÉRREZ CRUZ

DR. RICARDO RICO MOLINA



**MANUAL PARA PRÁCTICAS DEL LABORATORIO DE CÓMPUTO
PARA LA ASIGNATURA DE SISTEMAS OPERATIVOS.**

IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

Espacio académico: CENTRO UNIVERSITARIO NEZAHUALCÓYOTL								
Programa educativo INGENIERÍA EN SISTEMAS INTELIGENTES					Área de docencia: HERRAMIENTA PARA LOS SISTEMAS INTELIGENTES			
Aprobación de los HH Consejos Académico y de Gobierno			Fecha: SEPTIEMBRE 2018		Programa elaborado por: Yaroslav Aarón Albarrán Fernández, Doricela Gutiérrez Cruz y Carmen Liliana Rodríguez Páez			
Nombre de la unidad de aprendizaje: SISTEMAS OPERATIVOS					Fecha de elaboración: agosto 2018			
Clave	Horas de Teoría	Horas de Práctica	Total de horas	Créditos	Área curricular:	Carácter de la unidad de aprendizaje	Núcleo de formación	Modalidad
L40626	4	1	5	9	HERRAMIENTAS PARA LOS SISTEMAS INTELIGENTES	Obligatoria	SUSTANTIVA	ESCOLARIZADA CON ADMINISTRACIÓN FLEXIBLE DE LA ENSEÑANZA
Prerrequisitos (Conocimientos previos):			Unidad de aprendizaje antecedente:			Unidad de aprendizaje consecuente:		
PROGRAMACIÓN			NINGUNA			NINGUNA		
Programas en los que se imparte: LICENCIATURA DE INGENIERÍA EN SISTEMAS INTELIGENTES								

EL PRESENTE MANUAL DE PRÁCTICAS HA SIDO AVALADO EN EL MES DE SEPTIEMBRE DE 2018 POR:

EL PRESENTE MANUAL DE PRÁCTICAS HA SIDO AVALADO EN EL MES DE SEPTIEMBRE DE 2018 POR:



ÍNDICE

Directorio UAEM	4
Directorio del Centro Universitario Nezahualcóyotl	5
Ubicación de la asignatura de Algoritmos Genéticos, dentro del programa de la Lic. en Ing. en Sistemas Inteligentes.	6
Secuencia Didáctica	7
Práctica 1	
Introducción a los Sistemas Operativos	
Objetivo	8
Introducción	8
Desarrollo	10
Bibliografía	11
Práctica 2	
Procesos (Sistema)	12
Objetivo	12
Introducción	12
Desarrollo	15
Bibliografía	18
Práctica 3	
Procesos (POSIX)	19
Objetivo	19
Introducción	19
Desarrollo	20
Bibliografía	23
Práctica 4	
Terminal (Emulación)	24
Objetivo	24
Introducción	24
Desarrollo	25
Bibliografía	27
Práctica 5	
Terminal (Sistema Android)	28
Objetivo	28
Introducción	28
Desarrollo	29
Bibliografía	30
Práctica 6	
Gestión de Memoria	32
Objetivo	32
Introducción	32
Desarrollo	34
Bibliografía	36

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

DIRECTORIO

Dr. en Ed. Alfredo Barrera Baca

RECTOR

M. en S. P. María Estela Delgado Maya

SECRETARIA DE DOCENCIA

Dr. en C.I.Amb. Carlos Eduardo Barrera Díaz

SECRETARIO DE INVESTIGACIÓN Y ESTUDIOS AVANZADOS

Dr. en C.S. Luis Raúl Ortiz Ramírez

SECRETARIO DE RECTORÍA

Dr. en A. José Edgar Miranda Ortiz

SECRETARIO DE DIFUSIÓN CULTURAL

M. en Com. Jannet Socorro Valero Vilchis

SECRETARIA DE EXTENSIÓN Y VINCULACIÓN

M. en E. Javier González Martínez

SECRETARIO DE ADMINISTRACIÓN

Dr. en C.C. José Raymundo Marcial Romero

SECRETARIO DE PLANEACIÓN Y DESARROLLO INSTITUCIONAL

M. en L.A. María del Pilar Ampudia García

SECRETARIA DE COOPERACIÓN INTERNACIONAL

Dra. en C.S. y Pol. Gabriela Fuentes Reyes

ABOGADA GENERAL

Lic. en Com. Gastón Pedraza Muñoz

DIRECTOR GENERAL DE COMUNICACIÓN UNIVERSITARIA

M. en R.I. Jorge Bernaldez García

SECRETARIO TÉCNICO DE LA RECTORÍA

M. en A.P. Guadalupe Santamaría González

DIRECTORA GENERAL DE CENTROS UNIVERSITARIOS Y UAP

M. en A. Ignacio Gutiérrez Padilla

CONTRALOR UNIVERSITARIO

CENTRO UNIVERSITARIO NEZAHUALCÓYOTL

DIRECTORIO

*M. en D. Juan Carlos Medina Huicochea
Encargado del despacho de C.U. Nezahualcóyotl*

*M. en C. José Antonio Castillo Jiménez
Subdirector Académico*

*Lic. en E. Ramón Vital Hernández
Subdirector Administrativo*

*Dra. en C. S. María Luisa Quintero Soto
Coordinadora de Investigación y Estudios Avanzados*

*Lic. en A. E. Víctor Manuel Durán López
Coordinador de Planeación y Desarrollo Institucional*

*Dr. en R.I. Rafael Alberto Duran Gómez
Coordinador de la Licenciatura en Comercio Internacional*

*Mtra. Gabriela Kramer Bustos
Coordinadora de la Licenciatura en Educación para la Salud*

*Dra. Ricardo Rico Molina
Coordinador de la licenciatura en Ingeniería en Sistemas Inteligentes*

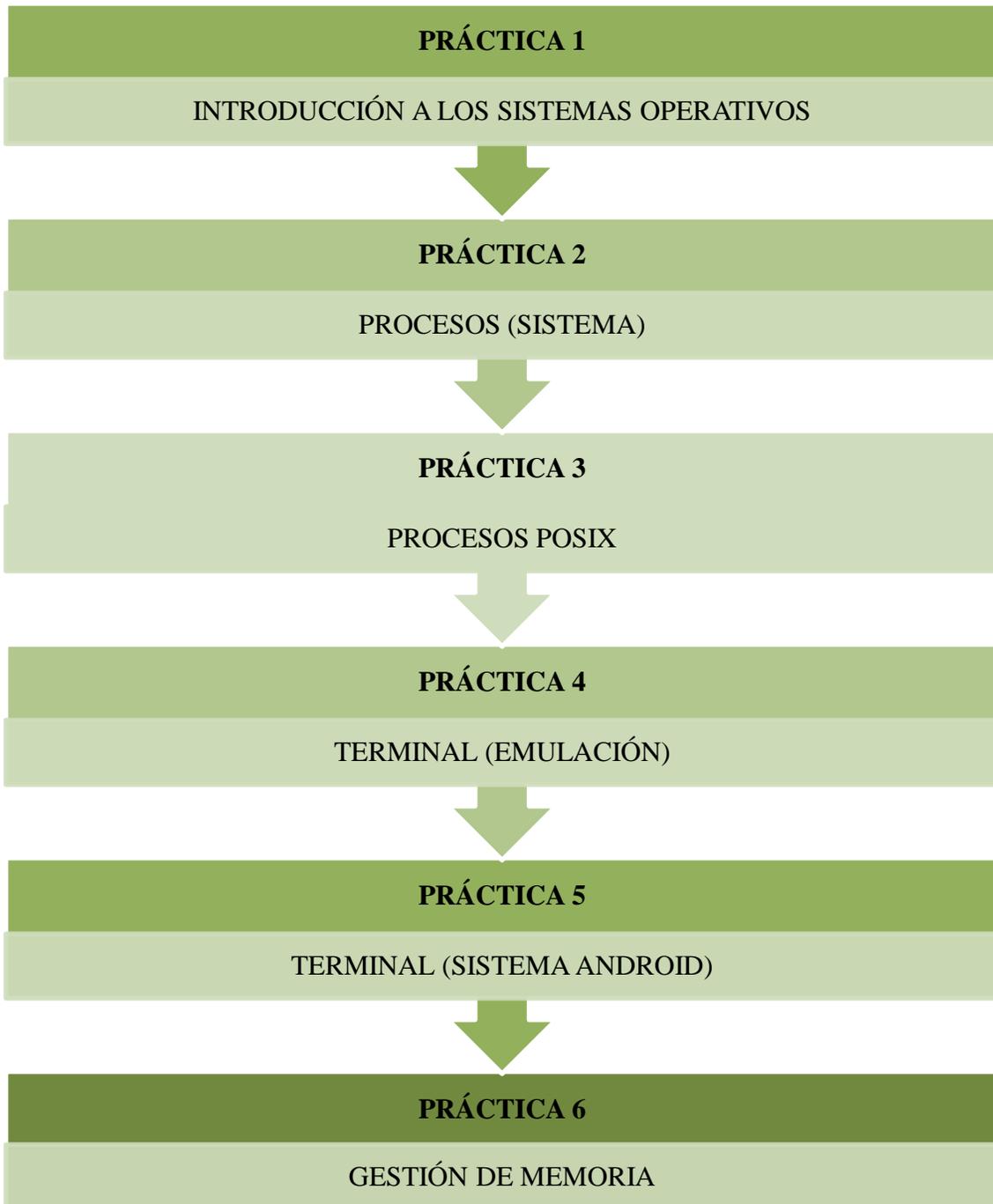
*D. En U. Noé Gaspar Sánchez
Coordinador de Ingeniería en Transporte*

*M. En C.C. Erick Nicolás Cabrera Álvarez
Coordinador de Licenciatura en Seguridad Ciudadana*

Ubicación de la asignatura de Sistemas Operativos, dentro del programa de la Lic. en Ing. en Sistemas Inteligentes.

SECUENCIA DIDÁCTICA

	PROGRAMACIÓN DE INGENIERÍA DE SOFTWARE	TRATAMIENTO DE LA INFORMACIÓN	SOFTWARE DE BASE	REDES	ARQUITECTURA DE COMPUTADORAS	MATEMÁTICAS	ENTORNO SOCIAL
	<p>PROGRAMACIÓN DE SISTEMAS DE SOFTWARE</p> <p>CONOCIMIENTO DE SISTEMAS DE SOFTWARE</p> <p>INGENIERÍA DE SISTEMAS</p> <p>CONOCIMIENTO DE SISTEMAS DE SOFTWARE</p> <p>PROGRAMACIÓN DE SISTEMAS DE SOFTWARE</p>	<p>PROGRAMACIÓN DE SISTEMAS DE SOFTWARE</p>	<p>PROGRAMACIÓN DE SISTEMAS DE SOFTWARE</p>	<p>PROGRAMACIÓN DE SISTEMAS DE SOFTWARE</p>	<p>PROGRAMACIÓN DE SISTEMAS DE SOFTWARE</p>	<p>MATEMÁTICAS</p>	<p>ENTORNO SOCIAL</p>



PRÁCTICA 1. INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

Objetivo:

El alumno conocerá los componentes que integran a un sistema operativo, así como los gestores que permiten mostrar la información del mismo.

Introducción:

La evolución del hardware ha sido constante: del monoprocesador a los multiprocesadores, grandes cantidades de memoria, incorporación de nuevos dispositivos, interconexión con otros sistemas ... pero ¿qué ha pasado con los sistemas operativos? Los sistemas operativos debieron acompañar la evolución. Conceptos comunes de lo que es un sistema operativo es que es un manejador de recursos, un programa de control, el programa que se ejecuta constantemente en una computadora, etc.

Tanenbaum lo expresa desde dos puntos de vista (complementarios): como máquina extendida o virtual más fácil de entender y programar que el hardware puro 1 y como administrador de recursos, entendiendo por “recursos” a los componentes tanto físicos como lógicos: el procesador, memoria, discos, ratones o archivos. (Ver Tanenbaum & Woodhull “Sistemas Operativos, diseño e implementación”).

Desarrollo:

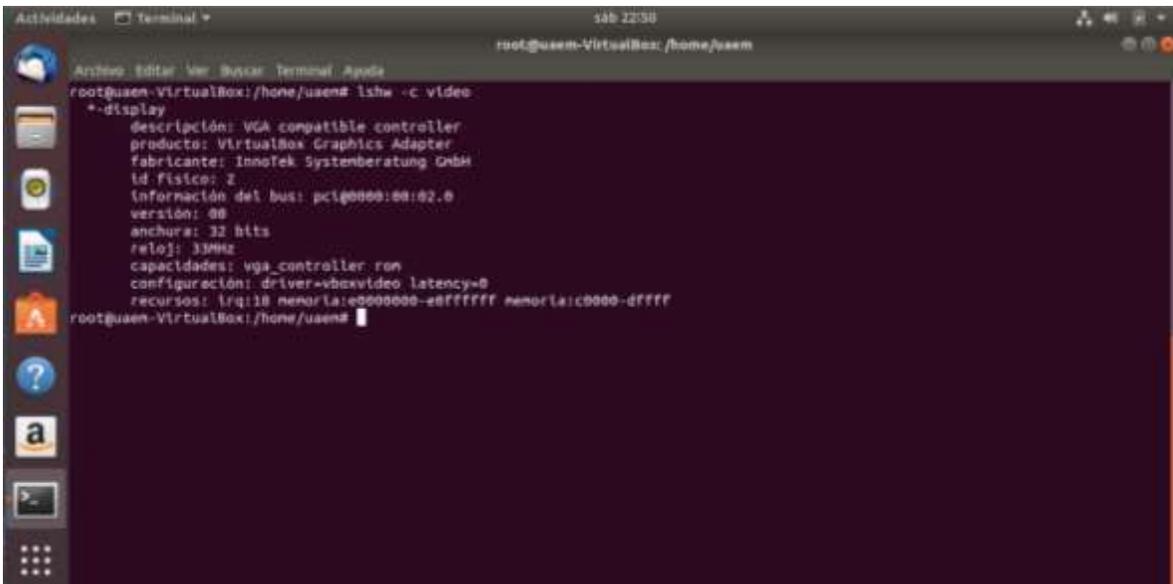
Como parte de los sistemas operativos es necesario saber información sobre el sistema, el conocer las características nos puede permitir hacer cambios o actualizaciones a nivel de hardware o para poder buscar algún driver en específico.

Para aquellos que empiezan a trabajar con Ubuntu es importante conocer las herramientas necesarias para tener la información del sistema y de las aplicaciones que gestionan su funcionamiento, así como la herramienta que regula la gestión de tareas.

La primera de las herramientas es sysinfo, el cual nos permite conocer las propiedades del sistema, para poder instalarlo, abrir una terminal y cambiar de usuario a root, una vez como super usuario escribir: **apt-get install sysinfo**, para poder tener acceso a la aplicación como root, se debe escribir lo siguiente; sysinfo &.



Esta herramienta es básica y te permite conocer aspectos generales del sistema. Otra de las herramientas que te puede dar información del sistema es lshw, la cual en algunas versiones ya viene agregada, pero de no ser así se debe instalar como la mencionada anteriormente, esta tiene la ventaja que genera la información de manera más específica, es una buena opción para obtener información de tu sistema.



Muestra información de dispositivos USB, PCI, tarjetas de red, discos, sistemas de fichero, entre otros y tiene una versión gráfica con lshw-gtk, esto depende de si trabajas con gnome o kde.

Como parte de las herramientas, **dmidecode**, es una herramienta más robusta y puedes consultar información del sistema, también tiene unos complementos para decodificaciones del DMI de la SMBIOS, para entendernos la SMBIOS define estructuras de datos en las que se almacena información de la BIOS (que a su vez contiene información de hardware muy valiosa).

```
root@uam-VirtualBox:~/home/uam# dmidecode processor-family
# dmidecode 3.1
Getting SMBIOS data from sysfs.
SMBIOS 2.5 present.
18 structures occupying 458 bytes.
Table at 0x00010000.

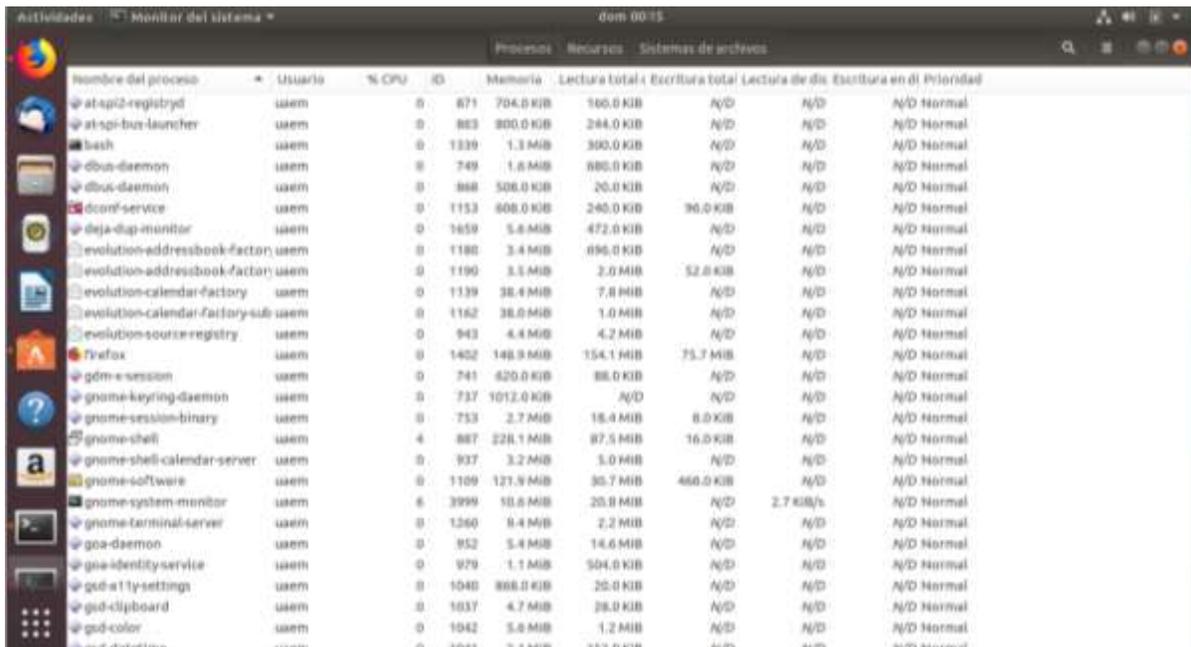
Handle 0x0000, DMI type 0, 28 bytes
BIOS Information
    Vendor: Innotek GmbH
    Version: VirtualBox
    Release Date: 17/01/2006
    Address: 0x00000
    Runtime Size: 128 kB
    ROM Size: 128 kB
    Characteristics:
        ISA is supported
        PCI is supported
        Boot from CD is supported
        Selectable boot is supported
        8042 keyboard services are supported (bit 06)
        CGA/mono video services are supported (bit 18b)
        ACPI is supported

Handle 0x0001, DMI type 1, 27 bytes
System Information
    Manufacturer: Innotek GmbH
    Product Name: VirtualBox
    Version: 1.2
    Serial Number: 8
    UUID: C4328368-1818-4825-80F3-C726F0E18E48
    Make-up Type: Power Switch
    SKU Number: Not Specified
    Family: Virtual Machine
```

Otra forma de obtener aplicaciones es por medio del centro de software, es de una manera más práctica, solo das doble click en el icono y escribes para nuestro caso “Monitor del sistema”.



Una vez instalado, se puede ejecutar desde terminal o se busca el icono creado en el menú de herramientas, esta herramienta la ventaja que tiene es que consta de una interfaz gráfica de usuario, lo cual permite tener no solo una visión de los procesos, sino también una serie de estadísticas del uso del sistema.



A continuación, se muestra un ejemplo de cómo esta estructurada su interfaz.



Para no dejar de lado a otras herramientas, nos encontramos con una que se hace parte de nuestro escritorio y nos muestra en tiempo real los cambios generados en los procesos, esta herramienta se llama conky, un widget que nos **muestra información** como por ejemplo la temperatura de nuestros procesadores, la capacidad de disco, el uso de la RAM, y muchas otras características más.



Una vez que el alumno instalo y conoció estas herramientas debe realizar las siguientes actividades:

Actividades que desarrollar:

1. El alumno deberá buscar tres herramientas que deberá instalar y hacer una descripción de su instalación, características y funcionamiento.
2. Hacer uso de las diversas variantes del comando ls y otros como:
lsusb
lsscsi
lsblk
lshw
top
free
fdisk
htop
Para este caso ejecutar y buscar más variantes relacionadas con este comando.
3. Hacer una comparativa entre los sistemas operativos de Windows y Linux en relación con los gestores de procesos y rendimiento del sistema.
4. El alumno tendrá que visualizar por medio de cat el contenido de las siguientes rutas y definir que tipo de información contienen:
/proc/filesystems
/proc/cpuinfo
/proc/interrupts
/proc/loadavg
5. Verificar herramientas que realicen funciones similares en el entorno de Windows.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

Bibliografía

1. Carretero, P. (2001). Sistemas operativos: Una visión aplicada. Primera Edición. Madrid, Editorial McGraw-Hill.
2. Deitel, M. (1993). Introducción a los sistemas operativos. Segunda Edición México, Editorial Addison Wesley Longman de México,.
3. Dhamdhere, D. (2008). Sistemas operativos. Segunda Edición. México, Editorial McGraw-Hill.
4. Flynn, M. (2001). Sistemas operativos. Tercera Edición. México, Editorial International Thomson.
5. Galli, D. (2000). Distributed operating systems. New Jersey, Editorial Prentice-Hall.
6. McIver McHoes, A. (2011). Sistemas operativos. Sexta Edición. México, Editorial CENGAGE Learning.
7. Pérez, F. (2003). Problemas de sistemas operativos: de la base al diseño. Segunda Edición. Madrid. Editorial McGraw-Hill.
8. Ramez, E. (2010). Operating Systems: A Spiral Approach. Primera Edición. México, Editorial McGraw-Hill.
9. Silberschatz, A. (2006). Fundamentos de sistemas operativos. Séptima Edición. Madrid, Editorial McGraw-Hill.
10. Silberschatz, A. (2008). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
11. Tanenbaum, A. S. (1995). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
12. Tanenbaum, A. (1996). Sistemas operativos distribuidos. México, Editorial Prentice Hall.
13. Tanenbaum, A. (1998) Sistemas operativos: Diseño e implementación. Segunda Edición México, Editorial Prentice Hall.

14. Tanenbaum, A. (2009). *Sistemas operativos modernos*. Tercera Edición. México, Editorial Prentice Hall.

15. Torres, J. (2001). *Conceptos de sistemas operativos: Teoría y práctica*. Primera Edición. México, Editorial Trillas.

PRÁCTICA 2

PROCESOS (SISTEMA).

Objetivo

El alumno conocerá el proceso que están relacionados con los sistemas operativos y su impacto en el sistema.

Introducción

“Un proceso es básicamente un entorno formado por todos los recursos necesarios para ejecutar programas. Desde el punto de vista del SO, un proceso es un objeto más que hay que gestionar y al cual hay que dar servicio”¹

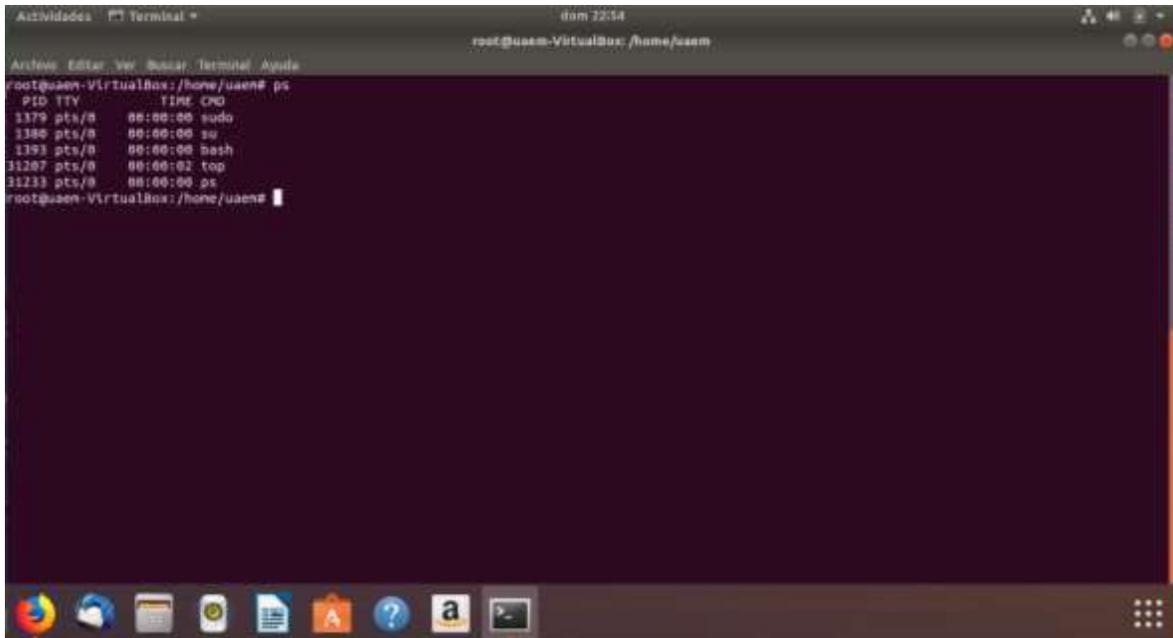
Es interesante notar que un programa como tal actúa de forma pasiva y su estructura la complementa un código que permite vincular las ordenes con el sistema, pero cuando este es ejecutado genera peticiones al sistema y entra el programa en acción, esto indica que los procesos son dinámicos, están en constante cambio debido a estos recursos necesarios, ya que al intentar realizar algún tipo de acción puede ser que tenga que permanecer a la espera de que dicho recurso esté disponible, por ejemplo una petición de lectura del disco duro, y que el brazo lector del disco duro lo esté utilizando otro proceso.

Desarrollo

Como parte del sistema se encuentran una serie de comandos que permiten administrar y gestionar los procesos, pero hay librerías de lenguaje c++, java entre otros, que permiten tener el control y la ejecución de los mismos, los clasificaremos primeramente los de terminal y después realizaremos una serie de ejemplos en código.

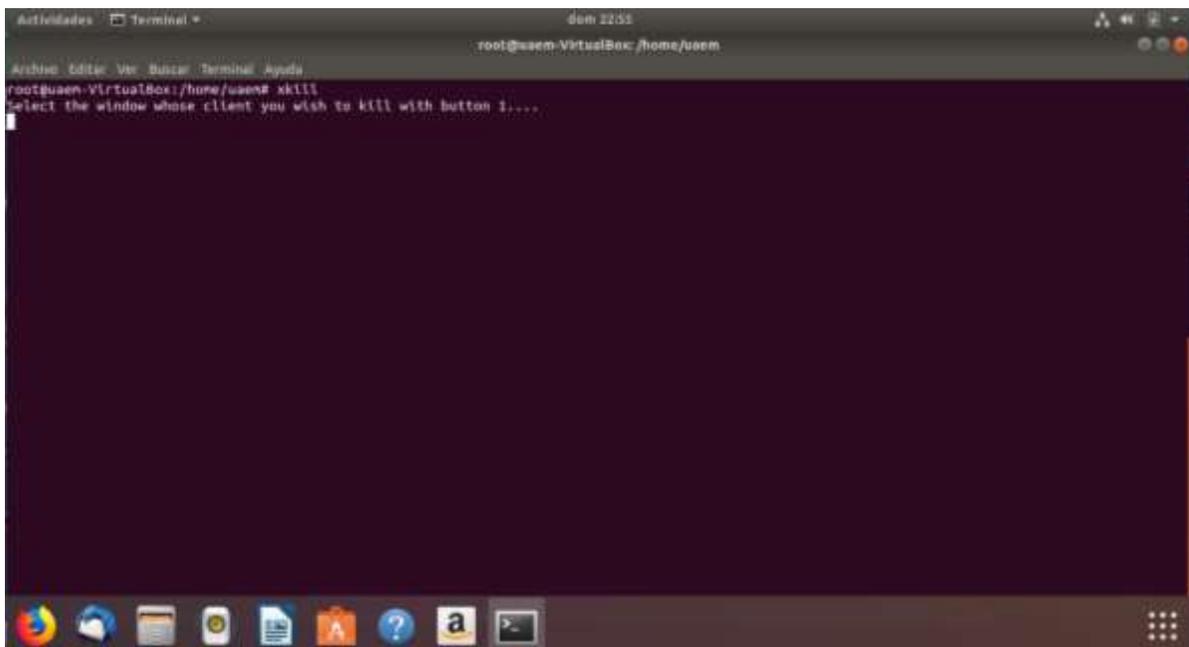
El comando top nos muestra los procesos en ejecución en tiempo real en nuestro sistema Ubuntu.

¹ Irene Rodil Jiménez y Camino Pardo de Vega (2010): Operaciones auxiliares con tecnologías de la información y la comunicación. (1ª Edición) Madrid: Paraninfo



```
root@uasm-VirtualBox: /home/uasm# ps
  PID TTY          TIME CMD
 3379 pts/0    00:00:00 sudo
 3380 pts/0    00:00:00 su
 3393 pts/0    00:00:00 bash
33207 pts/0    00:00:02 top
33233 pts/0    00:00:00 ps
root@uasm-VirtualBox: /home/uasm#
```

Una opción para la terminación de procesos es kill o killall, los cuales a través de su numero de proceso o el nombre de la aplicación actúan, pero hay un complemento que actúa en base a la selección de la aplicación que queremos detener y esa se llama xkill.



```
root@uasm-VirtualBox: /home/uasm# xkill
Select the window whose client you wish to kill with button 1....
```

1. Realizar la ejecución del comando kill y killall, que aplique para tres aplicaciones, una debe ser por el nombre y las otras por su número de proceso.
2. Ejecutar los siguientes comandos pidof, nice, renice, ps-aux y time, explicado su funcionamiento de cada uno de ellos, esto debe ser realizado en un script de bash.
3. Escribir las opciones de init para el manejo de procesos y ejemplificar su uso por medio de terminal.
4. Investigar sobre el comando de exec en terminal y realizar un ejemplo.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

Bibliografía

1. Carretero, P. (2001). Sistemas operativos: Una visión aplicada. Primera Edición. Madrid, Editorial McGraw-Hill.
2. Deitel, M. (1993). Introducción a los sistemas operativos. Segunda Edición México, Editorial Addison Wesley Longman de México,.
3. Dhamdhere, D. (2008). Sistemas operativos. Segunda Edición. México, Editorial McGraw-Hill.
4. Flynn, M. (2001). Sistemas operativos. Tercera Edición. México, Editorial International Thomson.
5. Galli, D. (2000). Distributed operating systems. New Jersey, Editorial Prentice-Hall.

6. McIver McHoes, A. (2011). Sistemas operativos. Sexta Edición. México, Editorial CENGAGE Learning.
7. Pérez, F. (2003). Problemas de sistemas operativos: de la base al diseño. Segunda Edición. Madrid. Editorial McGraw-Hill.
8. Ramez, E. (2010). Operating Systems: A Spiral Approach. Primera Edición. México, Editorial McGraw-Hill.
9. Silberschatz, A. (2006). Fundamentos de sistemas operativos. Séptima Edición. Madrid, Editorial McGraw-Hill.
10. Silberschatz, A. (2008). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
11. Tanenbaum, A. S. (1995). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
12. Tanenbaum, A. (1996). Sistemas operativos distribuidos. México, Editorial Prentice Hall.
13. Tanenbaum, A. (1998) Sistemas operativos: Diseño e implementación. Segunda Edición México, Editorial Prentice Hall.
14. Tanenbaum, A. (2009). Sistemas operativos modernos. Tercera Edición. México, Editorial Prentice Hall.
15. Torres, J. (2001). Conceptos de sistemas operativos: Teoría y práctica. Primera Edición. México, Editorial Trillas.

PRÁCTICA 3

PROCESOS (POSIX).

Objetivo

El alumno desarrollara programas donde vincule los comandos de bash del sistema y el uso de POSIX.

Introducción.

POSIX es el acrónimo de Portable Operating System Interface; la X viene de UNIX como seña de identidad de la API (Application Programming Interface, interfaz de programación de aplicaciones). Son una familia de estándares de llamadas al sistema operativo definidos por el IEEE (Institute of Electrical and Electronics Engineers, Instituto de Ingenieros Eléctricos y Electrónicos) y especificados formalmente en el IEEE 1003. Persiguen generalizar las interfaces de los sistemas operativos para que una misma aplicación pueda ejecutarse en distintas plataformas. Estos estándares surgieron de un proyecto de normalización de las API y describen un conjunto de interfaces de aplicación adaptables a una gran variedad de implementaciones de sistemas operativos.

Desarrollo.

Identificación de procesos.

Los servicios para gestión de procesos son parte de los servicios de POSIX, estos servicios se han agrupado en diversas categorías que son la identificación, creación de procesos y el termino, los procesos son identificados por `pid_t`, este debe ser de tipo entero, por los valores de los procesos, Los cuales se pueden visualizar en el siguiente código:

- Obtener la identificación del proceso: `pid_t getpid(void)`
- Obtener la identificación del proceso padre: `pid_t getppid(void)`
- Obtener la identificación del usuario real: `uid_t getuid(void)`
- Obtener la dirección del usuario efectivo: `uid_t geteuid(void)`
- Obtener el identificador del grupo real: `gid_t getgid(void)`
- Obtener el identificador del grupo efectivo: `gid_t getegid(void)`

Entorno de un proceso:

Existen las funciones de entorno de proceso estas tienen la característica que las variables se pasan al mismo al momento comenzar su ejecución, Trabaja con variables de entorno y son accesibles por variables externas (`environ`) que se declara de tipo apuntador **extern char **environ**, las funciones asociadas son:

- Obtener el valor de una variable de entorno: `char *getenv(const char *nombre)`

- Definir el entorno de un proceso: `char setenv (char **env)`

Gestión de procesos.

Para esta parte se puede crear un proceso en un sistema operativo esto se debe realizar por medio de `fork()`, esto se realiza por medio de la clonación de procesos mediante una solicitud, la solicitud de este proceso genera el padre, que es este a su vez el proceso hijo, `fork` esta estructurado de la siguiente forma: `pid_t fork()`.

Ejecutar un programa.

Esta parte está asociada a un servicio de posix llamado `exec`, tiene la característica de recibir parámetros de ejecución y un referente de ello es la asociación con comandos del sistema y no solo eso su relación de ejecución con procesos. En POSIX existe una serie de funciones cuyos prototipos se muestran a continuación: `execl()`, `execv()`, `execle()`, `execve()`, `execlp()` y `execvp()`.

1. Realizar un ejemplo en `c++` donde ejemplifiques cada uno de los métodos de identificación de procesos.
2. Hacer un programa donde apliques gestión de procesos, para la obtención de path del sistema.
3. Desarrollar diversos programas donde apliques los métodos de ejecución de un programa, vinculándolos con los comandos del sistema.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

Bibliografía:

1. Carretero, P. (2001). Sistemas operativos: Una visión aplicada. Primera Edición. Madrid, Editorial McGraw-Hill.

2. Deitel, M. (1993). Introducción a los sistemas operativos. Segunda Edición México, Editorial Addison Wesley Longman de México,.
3. Dhamdhere, D. (2008). Sistemas operativos. Segunda Edición. México, Editorial McGraw-Hill.
4. Flynn, M. (2001). Sistemas operativos. Tercera Edición. México, Editorial International Thomson.
5. Galli, D. (2000). Distributed operating systems. New Jersey, Editorial Prentice-Hall.
6. McIver McHoes, A. (2011). Sistemas operativos. Sexta Edición. México, Editorial CENGAGE Learning.
7. Pérez, F. (2003). Problemas de sistemas operativos: de la base al diseño. Segunda Edición. Madrid. Editorial McGraw-Hill.
8. Ramez, E. (2010). Operating Systems: A Spiral Approach. Primera Edición. México, Editorial McGraw-Hill.
9. Silberschatz, A. (2006). Fundamentos de sistemas operativos. Séptima Edición. Madrid, Editorial McGraw-Hill.
10. Silberschatz, A. (2008). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
11. Stallings, W. (1995). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
12. Tanenbaum, A. (1996). Sistemas operativos distribuidos. México, Editorial Prentice Hall.
13. Tanenbaum, A. (1998) Sistemas operativos: Diseño e implementación. Segunda Edición México, Editorial Prentice Hall.
14. Tanenbaum, A. (2009). Sistemas operativos modernos. Tercera Edición. México, Editorial Prentice Hall.
15. Torres, J. (2001). Conceptos de sistemas operativos: Teoría y práctica. Primera Edición. México, Editorial Trillas.

PRÁCTICA 4

TERMINAL (EMULACIÓN).

Objetivo

El alumno implementara el uso de una terminal en un sistema operativo de Windows y configurándolo para el uso de servicios.

Introducción.

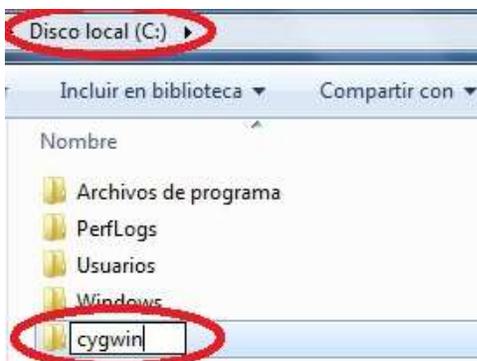
Cygwin es un programa que simula un ambiente “Unix” en el sistema operativo Windows.

Cygwin permite el uso de una terminal de comandos con algunas de las funcionalidades más comunes de los sistemas Unix/Linux mediante una biblioteca dinámica (cygwin1.dll) y algunas utilidades propias de los sistemas Unix (Para más información visiten: <http://www.cygwin.com>). El programa pre-configurado que utilizaremos (cygwin.exe) contiene la última versión de Cygwin e incluye varias herramientas para su mejor aprovechamiento, dentro de los cuales se incluye una muy importante: El “Servidor X”, para poder ejecutar aplicaciones gráficas nativas de Unix/Linux.

Desarrollo:

Configuración de Cygwin.

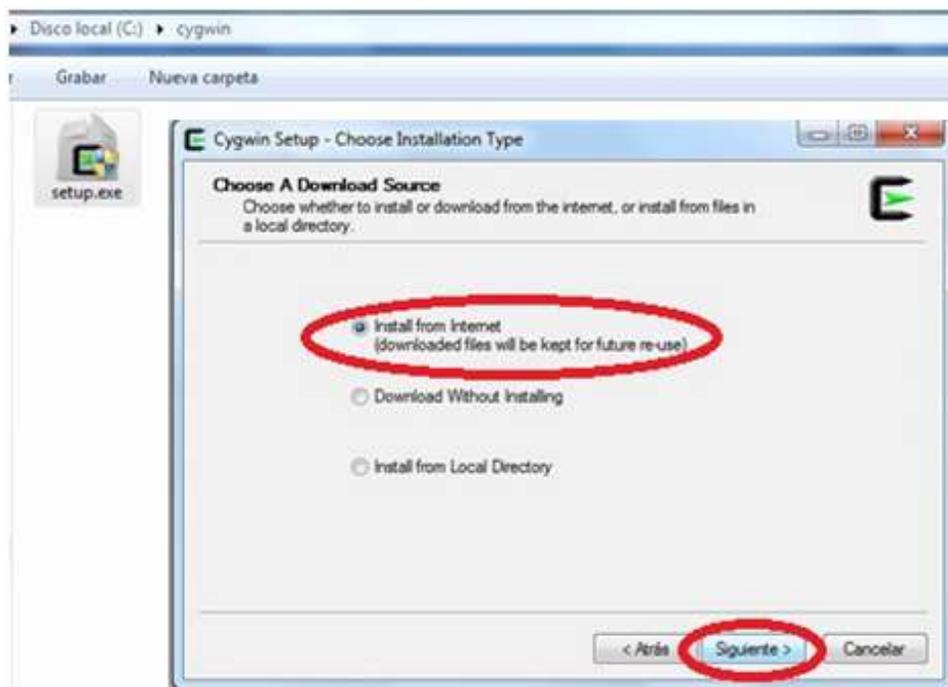
Descargamos el programa CygWin del siguiente Link: <http://cygwin.com/setup.exe>
En el disco local C: creamos una nueva carpeta llamada: **cygwin**.



Copiamos el Setup de CygWin en la carpeta que acabamos de crear e iniciamos la instalación.



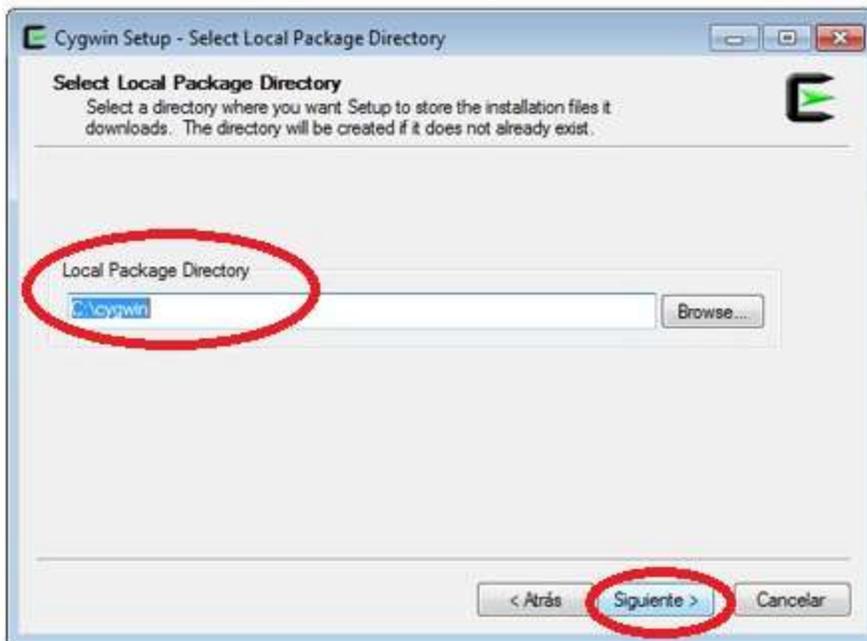
Seleccionamos la opción instalar desde internet.



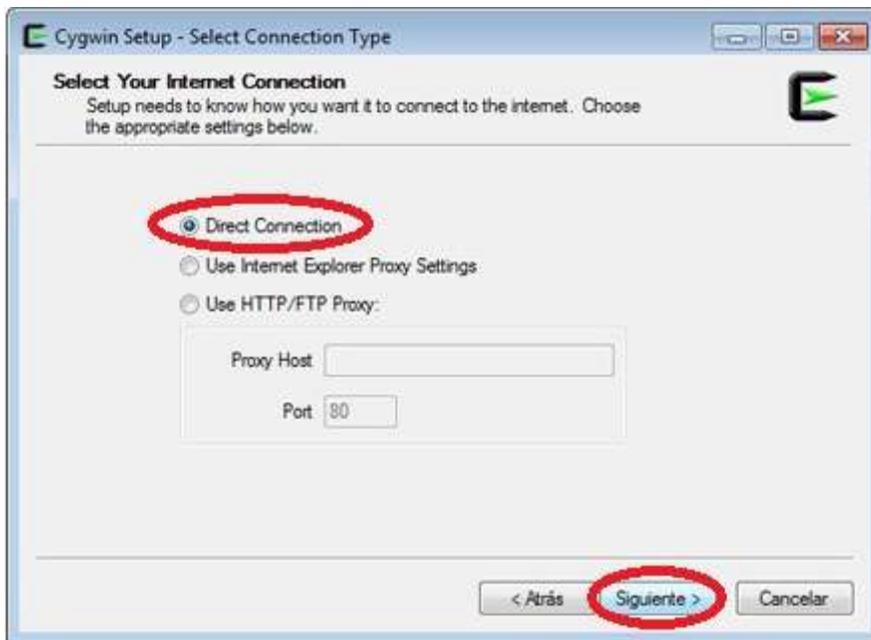
Verificamos que la Ruta de instalación sea la carpeta que acabamos de crear **C:\cygwin**



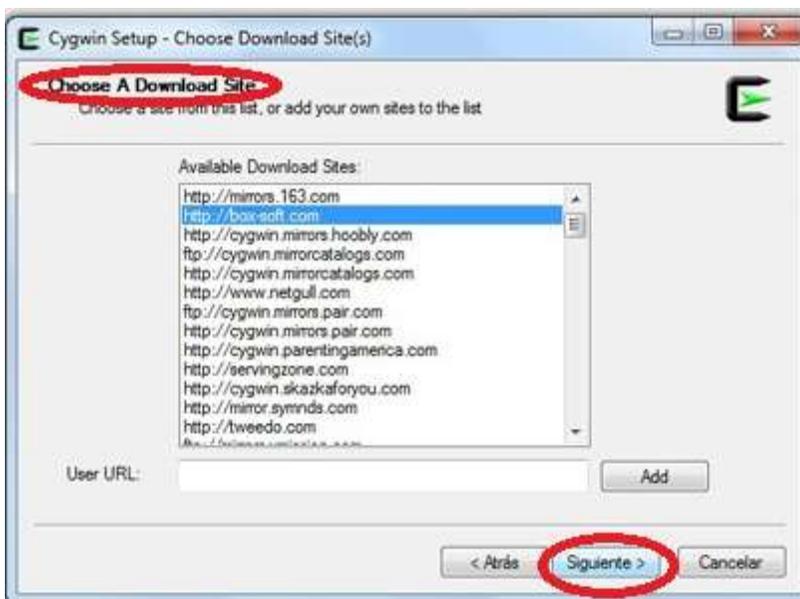
Acá también verificamos la Ruta C:\cygwin



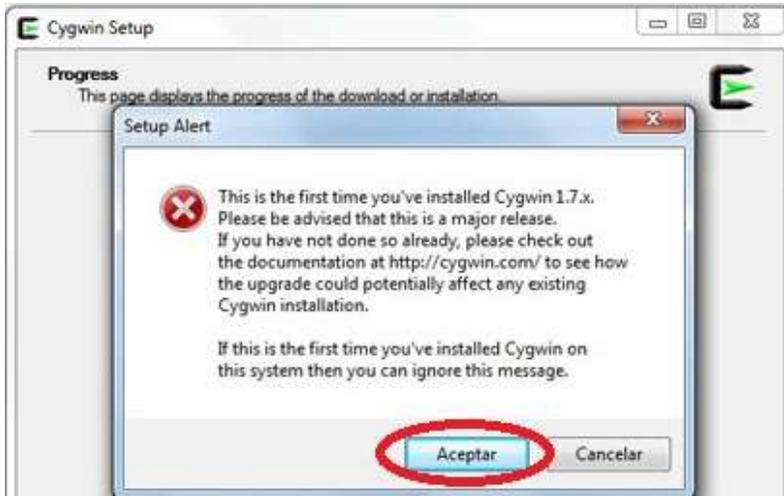
Seleccionamos la opción Conexión Directa.



Escogemos de la lista cualquier sitio para la descarga de nuestros archivos de instalación.



Les aparecerá un mensaje de alerta el cual simplemente nos dice que es la primera vez que se va a instalar CygWin.



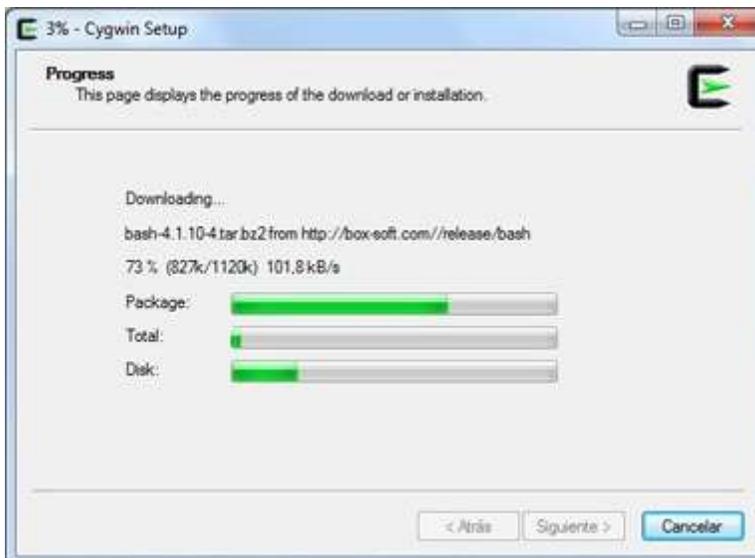
En el cuadro de búsqueda escribimos: **openssh** , damos click en el + de Net y escogemos la ultima versión **6.0p x-x** de nuestro programa cliente/servidor **OpenSSH**, también verificar que la casilla del cuadro este marcada con una **x** y click en siguiente:



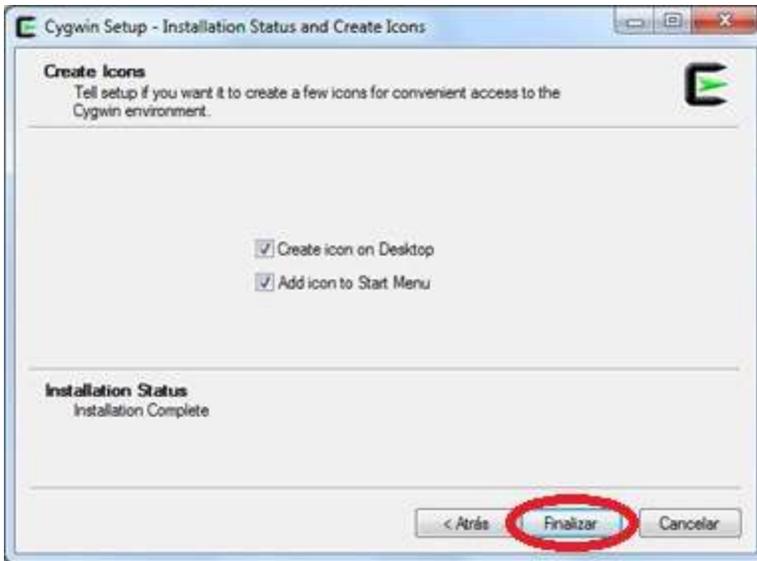
En esta ventana dejamos todo como esta y click en siguiente:



Inicia la descarga esto puede demorar un poco.

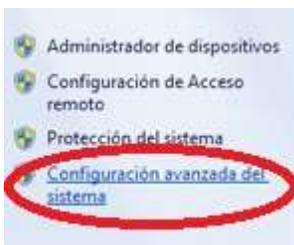


Finalizamos la instalación.



Agregamos la Ruta de CygWin en el PATH de Windows

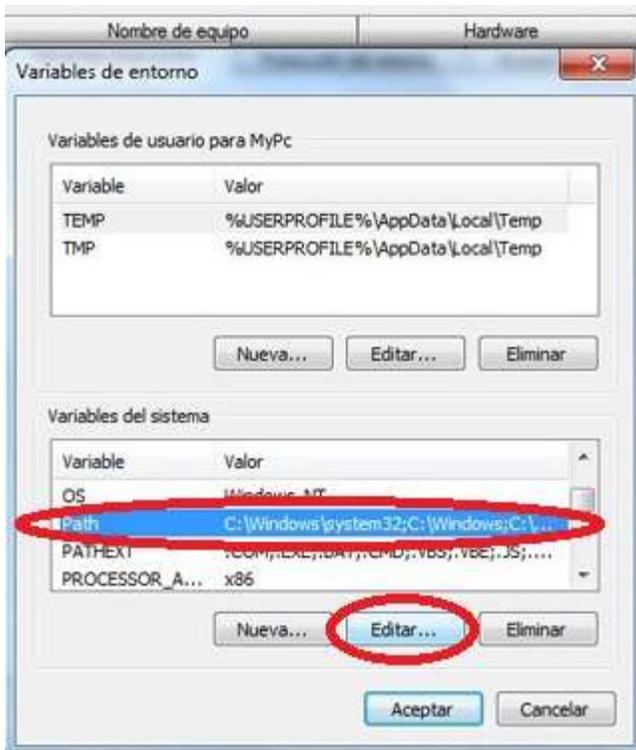
Nos vamos a **Propiedades de Equipo** y luego a configuración avanzada del sistema:



Click a **Variables de entorno**:



Buscamos la palabra **Path** y click en **Editar**

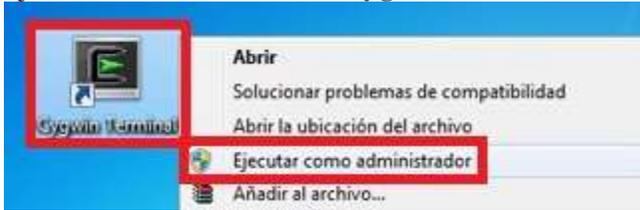


Al final del texto agregamos la ruta de la siguiente forma ;C:cygwin al final aceptamos todos los cambios.



Configuración de CYGWIN

Ejecutamos la terminal de **CygWin** como **Administrador**:



Ingresamos los siguientes comandos uno por uno, estos cambiaran los permisos de los archivos que vamos a usar y crearan el archivo sshd.log

```

chmod +r /etc/passwd
chmod u+w /etc/passwd
chmod +r /etc/group
chmod u+w /etc/group
chmod 755 /var
touch /var/log/sshd.log
chmod 664 /var/log/sshd.log
  
```

 A screenshot of a Cygwin terminal window. The terminal shows the following text:


```

Copying skeleton files.
These files are for the users to personalise their cygwin experience.
They will never be overwritten nor automatically updated.

'./.bashrc' -> '/home/Fabio/./.bashrc'
'./.bash_profile' -> '/home/Fabio/./.bash_profile'
'./.inputrc' -> '/home/Fabio/./.inputrc'
'./.profile' -> '/home/Fabio/./.profile'

Fabio@Fabio-HP ~
$ chmod +r /etc/passwd

Fabio@Fabio-HP ~
$ chmod u+w /etc/passwd

Fabio@Fabio-HP ~
$ chmod +r /etc/group

Fabio@Fabio-HP ~
$ chmod u+w /etc/group

Fabio@Fabio-HP ~
$ chmod 755 /var

Fabio@Fabio-HP ~
$ touch /var/log/sshd.log

Fabio@Fabio-HP ~
$ chmod 664 /var/log/sshd.log
  
```

Luego utilizamos el comando: **ssh-host-config** en seguida nos creara las diversas llaves que vamos a utilizar e iniciara una serie de preguntas:

```

$ ssh-host-config
*** Info: Generating /etc/ssh_host_key
*** Info: Generating /etc/ssh_host_rsa_key
*** Info: Generating /etc/ssh_host_dsa_key
*** Info: Generating /etc/ssh_host_ecdsa_key
*** Info: Creating default /etc/ssh_config file
*** Info: Creating default /etc/sshd_config file
*** Info: Privilege separation is set to yes by default since OpenSSH 3.3.
*** Info: However, this requires a non-privileged account called 'sshd'.
*** Info: For more info on privilege separation read /usr/share/doc/openssh/README.privsep.
*** Query: Should privilege separation be used? (yes/no) yes
*** Info: Note that creating a new user requires that the current account have
*** Info: Administrator privileges. Should this script attempt to create a
*** Query: new local account 'sshd'? (yes/no) yes
*** Info: Updating /etc/sshd_config file

*** Query: Do you want to install sshd as a service?
*** Query: (Say "no" if it is already installed as a service) (yes/no) yes
*** Query: Enter the value of CYGWIN for the daemon: [] ntsec tty
*** Info: On Windows Server 2003, Windows Vista, and above, the
*** Info: SYSTEM account cannot setuid to other users -- a capability
*** Info: sshd requires. You need to have or to create a privileged
*** Info: account. This script will help you do so.

*** Info: You appear to be running Windows XP 64bit, Windows 2003 Server,
*** Info: or later. On these systems, it's not possible to use the LocalSystem
*** Info: account for services that can change the user id without an
*** Info: explicit password (such as passwordless logins [e.g. public key
*** Info: authentication] via sshd).

*** Info: If you want to enable that functionality, it's required to create
*** Info: a new account with special privileges (unless a similar account
*** Info: already exists). This account is then used to run these special
*** Info: servers.

*** Info: Note that creating a new user requires that the current account
*** Info: have Administrator privileges itself.

*** Info: No privileged account could be found.

*** Info: This script plans to use 'cyg_server'.
*** Info: 'cyg_server' will only be used by registered services.
*** Query: Do you want to use a different name? (yes/no) yes
*** Query: Enter the new user name: sshd
*** Query: Reenter: sshd

*** Warning: Privileged account 'sshd' was specified,
*** Warning: but it does not have the necessary privileges.
*** Warning: Continuing, but will probably use a different account.
*** Warning: The specified account 'sshd' does not have the
*** Warning: required permissions or group memberships. This may
*** Warning: cause problems if not corrected; continuing...
*** Query: Please enter the password for user 'sshd':
*** Query: Reenter:

*** Info: The sshd service has been installed under the 'sshd'
*** Info: account. To start the service now, call 'net start sshd' or
*** Info: 'cygrunsrv -S sshd'. Otherwise, it will start automatically
*** Info: after the next reboot.

*** Info: Host configuration finished. Have fun!

```

Si se debe utilizar privilegios separados? **SI**
 Crear una nueva cuenta sshd? **SI**

Instalar ssh como servicio? **SI**

Valor para otra variable de entorno: **ntsec tty**

Usar un nombre diferente? **SI** (sshd, confirmamos el nombre sshd)

Crear contraseña: (123456, confirmamos la contraseña 123456)

***Mensaje "El Servicio SSHD ha sido instalado exitosamente"**

Por último habilitamos la autentificación LSA

cyglsa-config

*A la pregunta contestamos que **SI** y al final nos pide que Reiniciemos Windows.

```
$ cyglsa-config
Warning: Registering the Cygwin LSA authentication package requires
administrator privileges! You also have to reboot the machine to
activate the change.

Are you sure you want to continue? (yes/no) yes

Cygwin LSA authentication package registered.

Activating Cygwin's LSA authentication package requires to reboot.
```

Configuración línea de comandos de Windows

Abrimos la terminal de Windows presionamos las teclas **Win + R** escribimos **cmd** y copiamos los siguientes comandos:

cd C:\cygwin\bin

ash

/usr/bin/rebaseall

```
C:\Windows\system32>cd C:\cygwin\bin
C:\cygwin\bin>ash
"tty" option detected in CYGWIN environment variable.
CYGWIN=tty is no longer supported. Please remove it from your
CYGWIN environment variable and use a terminal emulator like mintty,
xterm, or rxvt.
$ /usr/bin/rebaseall
$
```

Iniciar el servicio:

Ejecutamos la terminal de CygWin como administrador y copiamos los siguientes comandos

chown system /etc/ssh*

chown system /var/empty

mkgroup -l > ..etcgroup

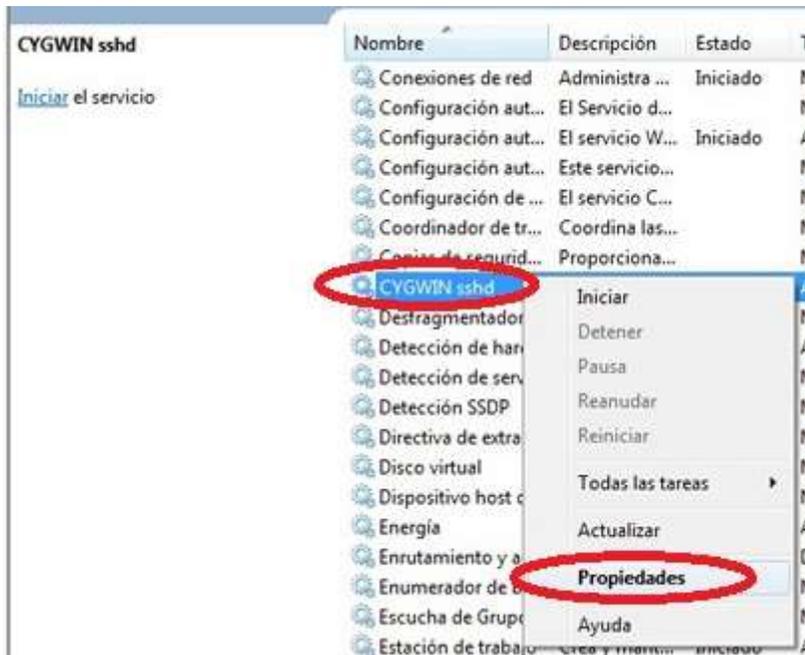
mkpasswd -l > ..etcpasswd

*Los comandos chown nos asegura que el servicio de cygwin no falle al iniciarse.

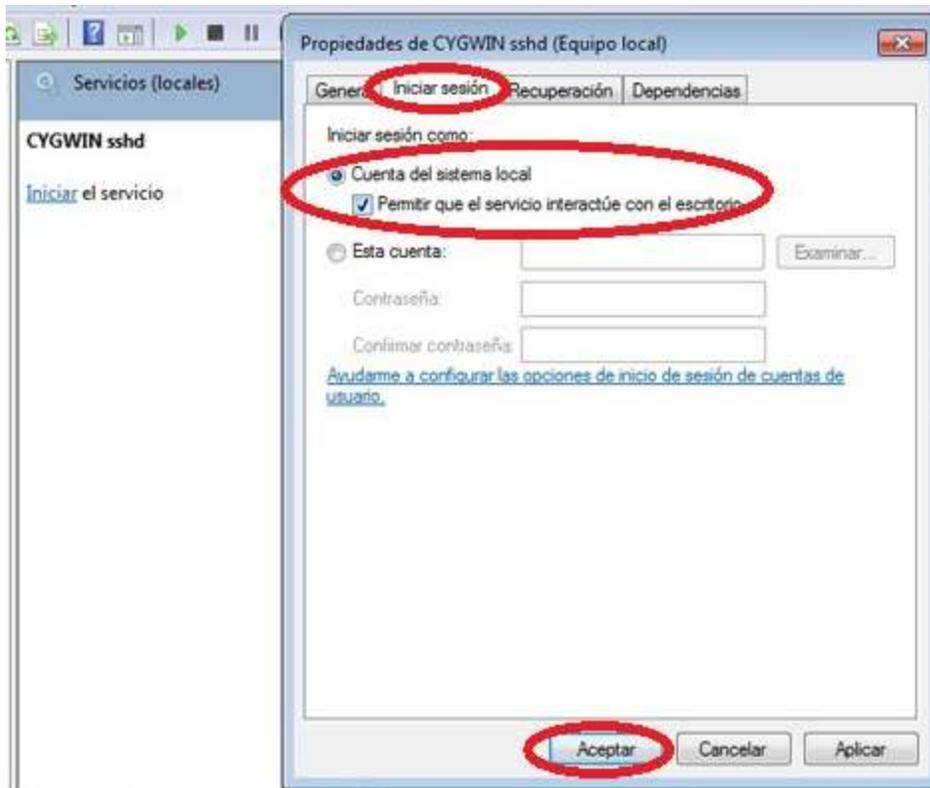
*mkgroup & mkpasswd coordina los servicios de usuario de cygwin con Windows para iniciar sesión.

* Al final cerramos la terminal.

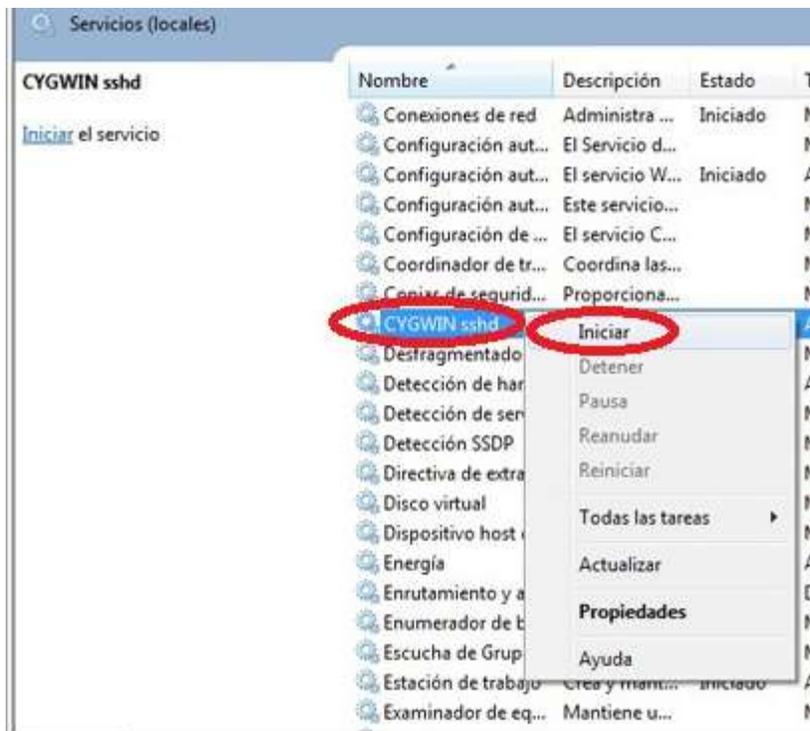
2. Por ultimo para asegurarnos de iniciar el servicio, presionamos nuevamente las teclas **Win + R** y escribimos: **services.msc** buscamos el servicio **CYGWIN sshd** y nos vamos a **propiedades**.



3. Click en la pestaña de **Iniciar Sesión** y marcamos las siguientes 2 opciones, aceptamos los cambios.

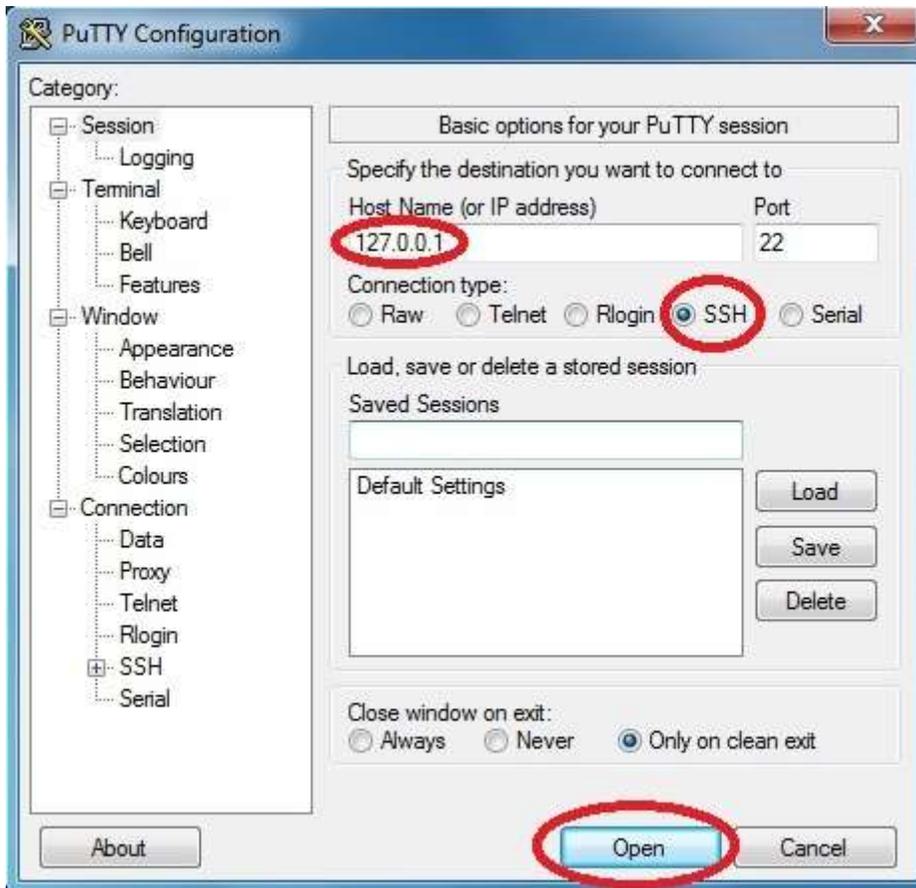


4. Por último iniciamos el servicio y ha quedado nuestro equipo como servidor.



Prueba del Servicio

Ejecutamos el programa no necesita instalación, y hacemos la prueba con localhost, con la dirección: **127.0.0.1** tener seleccionada la opción **SSH** y click en **open**.



En la terminal de PuTTY nos pedirá el nombre del usuario de la computadora servidor y su respectivo password correcto esto inicia la conexión cliente/servidor.

Ejercicios.

1. Realizar la implementación de tres servicios que se pueden implementar para Windows.
2. Ejecutar 10 comandos de sistema para ver la funcionalidad del software.
3. Buscar otras opciones de emular una terminal.
4. Generar un bash en una terminal online.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

Bibliografía:

1. Carretero, P. (2001). Sistemas operativos: Una visión aplicada. Primera Edición. Madrid, Editorial McGraw-Hill.
2. Deitel, M. (1993). Introducción a los sistemas operativos. Segunda Edición México, Editorial Addison Wesley Longman de México,.
3. Dhamdhere, D. (2008). Sistemas operativos. Segunda Edición. México, Editorial McGraw-Hill.
4. Flynn, M. (2001). Sistemas operativos. Tercera Edición. México, Editorial International Thomson.
5. Galli, D. (2000). Distributed operating systems. New Jersey, Editorial Prentice-Hall.
6. McIver McHoes, A. (2011). Sistemas operativos. Sexta Edición. México, Editorial CENGAGE Learning.
7. Pérez, F. (2003). Problemas de sistemas operativos: de la base al diseño. Segunda Edición. Madrid. Editorial McGraw-Hill.
8. Ramez, E. (2010). Operating Systems: A Spiral Approach. Primera Edición. México, Editorial McGraw-Hill.
9. Silberschatz, A. (2006). Fundamentos de sistemas operativos. Séptima Edición. Madrid, Editorial McGraw-Hill.
10. Silberschatz, A. (2008). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
- s, W. (1995). Sistemas operativos. Sexta Edición. México, Editorial Limusa.

12. Tanenbaum, A. (1996). *Sistemas operativos distribuidos*. México, Editorial Prentice Hall.

13. Tanenbaum, A. (1998) *Sistemas operativos: Diseño e implementación*. Segunda Edición México, Editorial Prentice Hall.

14. Tanenbaum, A. (2009). *Sistemas operativos modernos*. Tercera Edición. México, Editorial Prentice Hall.

15. Torres, J. (2001). *Conceptos de sistemas operativos: Teoría y práctica*. Primera Edición. México, Editorial Trillas.

PRÁCTICA 5

TERMINAL (SISTEMAS ANDROID).

Objetivo

El alumno implementara el uso de una terminal en un sistema operativo de Windows y configurándolo para el uso de servicios.

Introducción.

Una aplicación que ha mejorado el uso de una terminal en sistemas Android es *Termux*, que es un emulador de terminal para Android, que funciona directamente sin necesidad de rootear o realizar complicadas configuraciones, una ventaja es que se pueden agregar servicios de comunicación y de sistema.

Al instalar *Termux* en tu dispositivo móvil, dispondrás de un emulador de terminal para Android con un sistema *base mínimo*, pero que es fácilmente ampliable mediante el gestor de paquetes APT. Este gestor de paquetes es el que utilizan las distribuciones derivadas de Debian. Con lo que fácilmente podrás instalar los paquetes más habituales en este emulador de terminal para Android.

Desarrollo:

Una ventaja importante de esta terminal es que para los desarrolladores de aplicaciones es muy útil, ya que tiene una gama de lenguajes para poder usar.

Dentro de los más comunes se encuentra C, el cual necesita el apk (clang), también trabaja con el paquete: clang que es una interfaz de lenguaje C para LLVM, RE que es un compilador de lenguaje de programación D, construido con LLVM.

Otros de los lenguajes de uso por desarrolladores que tienen bastante importancia son: Haskell, Java, ECJ (Compilador de Eclipse), OpenJDK, Ceceo, OCaml, Octave, PHP, Picolisp, Python.

La interfaz es amigable, en caso de que se tenga problemas con el teclado que tiene integrado, se pueden adaptar otras opciones.

Una vez instalado *Termux*, dispondrás de un sistema básico que consiste en el gestor de paquetes APT y la colección de herramientas *busybox*. Puedes instalar más paquetes utilizando el gestor de paquetes.

Algunas de las funciones básicas son:

apt update. Actualiza la lista de paquetes disponibles.

apt search [paquete]. Permite buscar entre los diferentes paquetes que hay disponibles.

apt install [paquete] . Para instalar un paquete.

apt upgrade. Para actualizar todos los paquetes que estén desfasados, y que dispongan de una versión más actualizada de la que tenemos instalada.

apt show [paquete]. Muestra información relevante sobre un paquete.

apt list. Lista todos los paquetes disponibles.

apt list –installed . Lista todos los paquetes instalados en el sistema.

apt remove [paquete]. Permite quitar un paquete instalado en el sistema.

A continuación, se muestran algunas pantallas de su interfaz gráfica².



```
$ uname -a
Linux localhost 3.10.40-geec2459 #1 SMP PREEMPT Wed
Jan 28 22:14:35 UTC 2015 armv7l Android
$ apt update
Hit http://apt.termux.com stable InRelease
Hit http://apt.termux.com stable/main all Packages
Hit http://apt.termux.com stable/main arm Packages
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
$
```

```
# This is a comment
class Person
  attr_accessor :name

  def initialize(attributes = {})
    @name = attributes[:name]
  end

  def self.greet
    "hello"
  end
end

person1 = Person.new(:name => "Chris")
print Person::greet, " ", person1.name, "\n"
puts "another #{Person::greet} #{person1.name}"
```

[0] 0:bash* localhost 17:13 30 Jun-15

² Información obtenida de las siguientes fuentes.

https://play.google.com/store/apps/details?id=com.termux&hl=es_MX

<https://www.atareao.es/software/utilidades/terminal-para-android/>

```
12:00
$ apt install python
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
 python
0 upgraded, 1 newly installed, 0 to remove and 0 not
not upgraded.
Need to get 0 B/5147 kB of archives.
After this operation, 21.0 MB of additional disk s
pace will be used.
Selecting previously unselected package python.
(Reading database ... 6905 files and directories c
urrently installed.)
Preparing to unpack .../archives/python_3.4.3_i686
.deb ...
Unpacking python (3.4.3) ...
Setting up python (3.4.3) ...
$ python
Python 3.4.3 (default, Apr 5 2015, 00:59:51)
[GCC 4.8] on linux
Type "help", "copyright", "credits" or "license" f
or more information.
>>> import math
>>> math.pi
3.1415926535897931
>>> █
```

Ejercicios:

1. Instalar la terminal y ejecutar comandos del sistema para gestión de memoria.
2. Ejecutar bash dentro de la terminal de Android.
3. Buscar otras terminales y hacer una tabla donde muestres las ventajas y desventajas de esta terminal con respecto a las que investigaste.
4. Realiza pruebas con dos entornos de programación dentro de la terminal.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

Bibliografía:

1. Carretero, P. (2001). Sistemas operativos: Una visión aplicada. Primera Edición. Madrid, Editorial McGraw-Hill.
2. Deitel, M. (1993). Introducción a los sistemas operativos. Segunda Edición México, Editorial Addison Wesley Longman de México,.
3. Dhamdhere, D. (2008). Sistemas operativos. Segunda Edición. México, Editorial McGraw-Hill.
4. Flynn, M. (2001). Sistemas operativos. Tercera Edición. México, Editorial International Thomson.
5. Galli, D. (2000). Distributed operating systems. New Jersey, Editorial Prentice-Hall.
6. McIver McHoes, A. (2011). Sistemas operativos. Sexta Edición. México, Editorial CENGAGE Learning.
7. Pérez, F. (2003). Problemas de sistemas operativos: de la base al diseño. Segunda Edición. Madrid. Editorial McGraw-Hill.
8. Ramez, E. (2010). Operating Systems: A Spiral Approach. Primera Edición. México, Editorial McGraw-Hill.
9. Silberschatz, A. (2006). Fundamentos de sistemas operativos. Séptima Edición. Madrid, Editorial McGraw-Hill.
10. Silberschatz, A. (2008). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
11. Tanenbaum, A. S. W. (1995). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
12. Tanenbaum, A. (1996). Sistemas operativos distribuidos. México, Editorial Prentice Hall.
13. Tanenbaum, A. (1998) Sistemas operativos: Diseño e implementación. Segunda Edición México, Editorial Prentice Hall.
14. Tanenbaum, A. (2009). Sistemas operativos modernos. Tercera Edición. México, Editorial Prentice Hall.

15. Torres, J. (2001). Conceptos de sistemas operativos: Teoría y práctica. Primera Edición. México, Editorial Trillas.

PRÁCTICA 6.

GESTIÓN DE MEMORIA.

Objetivo.

El alumno conocerá los conceptos relacionados con el manejo de memoria, comandos relacionados y funciones en programación.

Introducción.

La forma en la que trabaja la memoria es tratar de conseguir que varios procesos puedan ejecutarse de forma concurrente, evitando los conflictos de uso, además es importante notar que también trata de proteger al sistema operativo y una de sus principales funciones es la de aprovechar eficazmente el espacio disponible: Minimizar la memoria para optimizar su uso, evitar en sistemas open source la fragmentación, la asignación adecuada de memoria a los diversos procesos, mejorar el rendimiento de los sistemas y haciéndolo en un tiempo de respuesta factible.

Desarrollo.

Como parte de el set de comandos del sistema, algunos de ellos generan información del sistema, como el comando Free en Linux **muestra la cantidad de memoria libre y usada que tiene el sistema**. Por una parte muestra la memoria física y por otra la swap, también muestra la memoria caché y de buffer consumida por el Kernel.



```
root@uaem-VirtualBox: /home/uaem
Archivo Editar Ver Buscar Terminal Ayuda
root@uaem-VirtualBox: /home/uaem# free -n
total usado libre compartido búfer/caché disponible
Memoria: 1921 1512 181 52 226 342
Swap: 472 47 424
```

Puede utilizar el comando vmstat para informar las estadísticas de memoria virtual y proporcionar información sobre eventos del sistema, como carga de CPU, paginación, número de cambios de contexto, interrupciones de dispositivo y llamadas del sistema. El comando vmstat también puede mostrar las estadísticas de intercambio, vaciado de memoria caché e interrupciones³.

³ https://docs.oracle.com/cd/E38897_01/html/E23086/spmonitor-22.html

```

root@uaem-VirtualBox: /home/uaem
Archivo Editar Ver Buscar Terminal Ayuda
root@uaem-VirtualBox: /home/uaem# vmstat
procs -----memoria----- ---swap-- -----lo----- -sistema-- -----cpu---
--
 r  b  swpd  libre búfer caché  st  so  bl  bo  in  cs  us  sy  id  wa  st
 2  0  49108  73916  10864 314120  0  2  73  34  231  768  29  3  67  1  0
root@uaem-VirtualBox: /home/uaem#
    
```

La siguiente tabla describe los campos del resultado del comando vmstat.

Otras opciones de visualización son : visualizar estadísticas de la memoria virtual, estadísticas de intercambio, eventos en el sistema e interrupciones de dispositivos.

```

root@uaem-VirtualBox: /home/uaem
Archivo Editar Ver Buscar Terminal Ayuda
root@uaem-VirtualBox: /home/uaem# vmstat
procs -----memoria----- ---swap-- -----lo----- -sistema-- -----cpu---
--
 r  b  swpd  libre búfer caché  st  so  bl  bo  in  cs  us  sy  id  wa  st
 2  0  49108  87708  12016 306332  0  2  72  34  230  763  29  3  67  1  0
root@uaem-VirtualBox: /home/uaem#
    
```

Una forma también de obtener información del sistema es por medio de java, a continuación se presentan dos ejemplos:

```

public class Memoria {
public static void main(String[] args) {
    String nombre = "os.name";
    String version = "os.version";
    String arquitectura = "os.arch";

    System.out.println("Nombre: " + System.getProperty(nombre));
    System.out.println("Version: " + System.getProperty(version));
    System.out.println("Arquitectura: " + System.getProperty(arquitectura));
    }
}
    
```

```
import java.util.Iterator;
import java.util.Map;
import java.util.Set;

public class Memoria {
    public static void main(String[] args) {
        Map mapa = System.getenv();

        Set keys = mapa.keySet();
        Iterator iterador = keys.iterator();
        while (iterador.hasNext()) {
            String llave = (String) iterador.next();
            String valor = (String) mapa.get(llave);

            System.out.println(llave + " = " + valor);
        }
    }
}
```

Ejercicios.

1. Investiga que comandos en Linux me permiten obtener información de las particiones y de la estructura de disco del sistema.
2. Desarrolla en investiga que otros métodos en java a parte de los trabajados en la práctica, me permiten obtener información del sistema.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

Bibliografía:

1. Carretero, P. (2001). Sistemas operativos: Una visión aplicada. Primera Edición.Madrid, Editorial McGraw-Hill.

2. Deitel, M. (1993). Introducción a los sistemas operativos. Segunda Edición México, Editorial Addison Wesley Longman de México,.
3. Dhamdhere, D. (2008). Sistemas operativos. Segunda Edición. México, Editorial McGraw-Hill.
4. Flynn, M. (2001). Sistemas operativos. Tercera Edición. México, Editorial International Thomson.
5. Galli, D. (2000). Distributed operating systems. New Jersey, Editorial Prentice-Hall.
6. McIver McHoes, A. (2011). Sistemas operativos. Sexta Edición. México, Editorial CENGAGE Learning.
7. Pérez, F. (2003). Problemas de sistemas operativos: de la base al diseño. Segunda Edición. Madrid. Editorial McGraw-Hill.
8. Ramez, E. (2010). Operating Systems: A Spiral Approach. Primera Edición. México, Editorial McGraw-Hill.
9. Silberschatz, A. (2006). Fundamentos de sistemas operativos. Séptima Edición. Madrid, Editorial McGraw-Hill.
10. Silberschatz, A. (2008). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
11. Stallings, W. (1995). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
12. Tanenbaum, A. (1996). Sistemas operativos distribuidos. México, Editorial Prentice Hall.
13. Tanenbaum, A. (1998) Sistemas operativos: Diseño e implementación. Segunda Edición México, Editorial Prentice Hall.
14. Tanenbaum, A. (2009). Sistemas operativos modernos. Tercera Edición. México, Editorial Prentice Hall.
15. Torres, J. (2001). Conceptos de sistemas operativos: Teoría y práctica. Primera Edición. México, Editorial Trillas.

